



Reactor

From complexity to clarity: demystifying agent development

Cihan Cinar

Software & AI Architect





Reactor

Cihan Cinar

Software & AI Architect

Givaudan

Linkedin: @cihancinar

Github: @cihancinar

X: @Cihan_Cinar_

<https://blog.cihan-cinar.com/>



Agentic AI Terms You Should Know!

Created By: Aishwarya Naresh Reganti

LLM



AI model that generates new data, generally used in tasks like text or image generation.

Reasoning



AI models analyzing tasks, selecting tools, and planning multi-step actions based on context

LRM



Large Reasoning Models: optimized for multi-step logical reasoning beyond just generation.

Continual Pretraining



A process where AI models are trained on new unlabeled data to improve performance on specific tasks.

Supervised Fine-Tuning



A process where AI models are trained on new labeled data to improve performance on specific tasks.

Distillation



A technique where a smaller AI model learns from a larger ones predictions instead of regular training

MoE



A neural network model that routes inputs to specialized sub-models for efficiency and better task handling.

HITL



Human-in-the-loop, a design where human oversight is integrated into AI decision-making

Reinforcement Learning



Training models through trial and error, with rewards/penalties based on its generated outputs

RLHF



Reinforcement Learning from Human Feedback. Human feedback is used as reward/penalty for model

Alignment



Final stage of AI training, done to align with human values & safety standards

Post-Training



Same as alignment, generally used to refer to and fine-tuning done after initial model training

Agentic AI



AI systems capable of making independent decisions and interacting with external tools.

Workflows



AI systems where models and tools are orchestrated through predefined code paths.

Agents



AI systems where models dynamically direct their own processes and tool usage

Multi-Agents



Systems where multiple AI agents collaborate to complete tasks by dividing responsibilities.

Design Patterns



Architectures and task breakdown strategies for designing effective AI agent systems

Agent Memory



The ability of an AI agent to retain and recall past information to improve task execution.

Short-Term Memory



Temporary storage that allows AI to remember information within a single session.

Long-Term Memory



Persistent storage that allows AI to retain knowledge across different sessions.

Procedural Memory



A form of long-term memory that enables AI to recall how to perform tasks.

Cognitive Architecture



The flow of code/prompts/LLM calls that takes user input and performs actions or generation

RAG



Retrieval-Augmented Generation. Appending retrieved information to improve model response

Agentic RAG



A RAG system where the agent actively decides when and how to retrieve information.

Tools



External functions or APIs that AI agents can call to perform specialized actions like calculations

Function Calling



A mechanism where AI agents invoke external functions dynamically based on task.

MCP



A standard for connecting AI systems to the business tools, data and external environments (by Anthropic)

Structured Outputs



A constraint-based approach where AI responses must follow predefined formats/types,

CoT



A reasoning method where the AI explicitly breaks down its steps to improve performance

Test-Time Scaling



techniques used to dynamically adjust an AI model's reasoning depth during inference

ReAct



A method where AI first reasons about a problem before selecting an appropriate action.

Reflection



A technique where AI reviews past decisions and refines future actions for improved performance.

Self-Healing



The ability of an AI agent to detect and correct its own mistakes without human intervention.

LLM Judge



A separate model used to assess and score outputs from another model, ensuring quality control.

Hybrid Models



AI models that use test-time scaling adjust reasoning depth dynamically based on task complexity

Chaining



An AI design pattern where AI breaks tasks into sequential steps, ensuring structured execution.

Routing



An AI design pattern where tasks are directed to different models or agents based on context.

Orchestrator



A control system that manages multiple agents, determining their roles and execution order.

Vertical Agents



AI agents specialized in a specific industry or domain, optimizing performance for targeted applications.

Overthinking



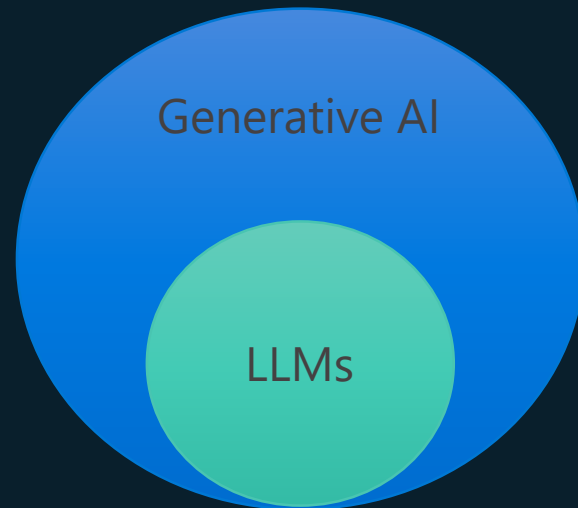
A common issue in LRMs where they overconsume time and tokens for every task—test-time scaling helps optimize this

What is Generative AI

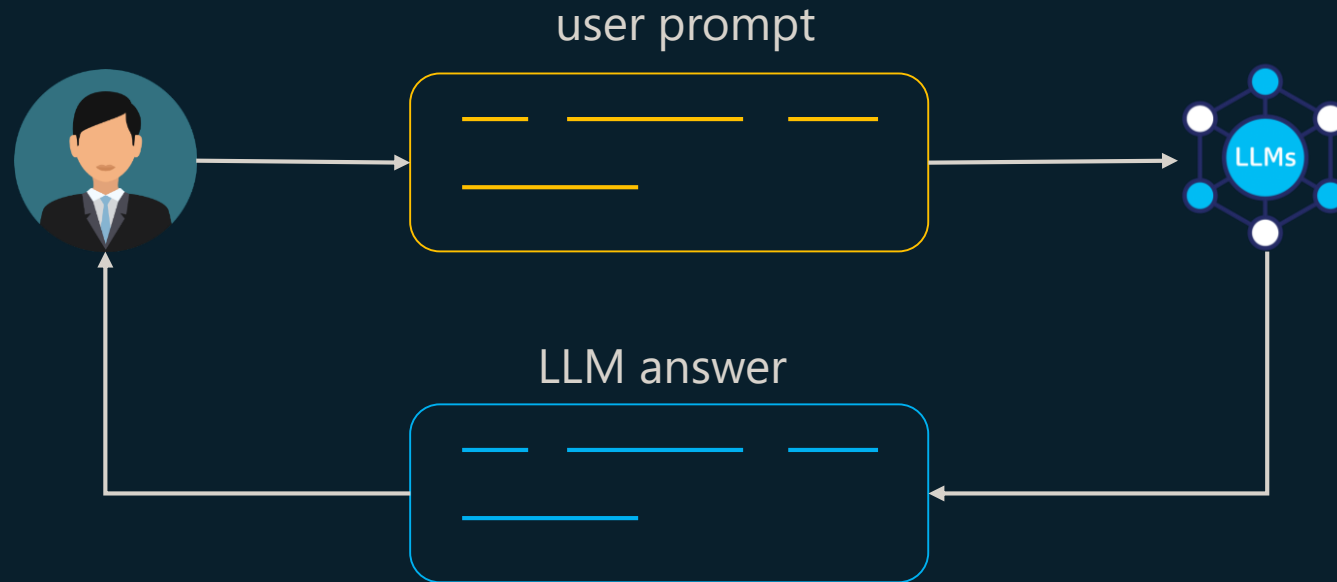
Generative AI is a branch of artificial intelligence focused on creating new content. Unlike traditional AI, which might classify or predict based on data, generative AI can produce original outputs such as images, text, music and more based on the vast amounts of data that the generative model was trained on.

What are large language models (LLMs)

Large language models (LLMs) are advanced AI systems that understand and generate natural language, or human-like text, using the data they've been trained on through machine learning techniques.



Chat with LLM



Limited to the LLMs knowledge and user input

Elements of a Good Prompt

Act as a helpful tutor who breaks down complex subjects into easy explanations.

I want you to explain the process of photosynthesis to a

14 year old student, to assist with biology exam preparations.

Your answer should be 300 words, written in a tone that's friendly and educational.

Persona: Ask the tool to take a role

Objective: What do you want the AI to do

Audience: Specify who it's for

Context: What does the tool need to know

Boundaries: Set your own direction & limitation

Tip 1

Give Clear Instructions

Use commands that instruct the AI tool on what you want to generate, such as 'explain', 'translate', 'summarize' or 'compare'.

Tip 2

Provide Context

Adding context and background information can help the tool to understand the task better. For example, mention the project type such as 'short story', 'report' or 'outline'

Tip 3

Iterate & Experiment

Try different instructions and techniques if you don't get the results you want. Prompting can be like an experiment that may require several rounds of iterations!

Chat Completion

- conversation-in and message-out
- LLM based on token

Tokens	Characters
3	18
Hello from Reactor	

- Context window
 - 4k input & 4k output tokens for common models
 - 128k+ input & 16k+ output tokens for newer models

Chat Completion

Request

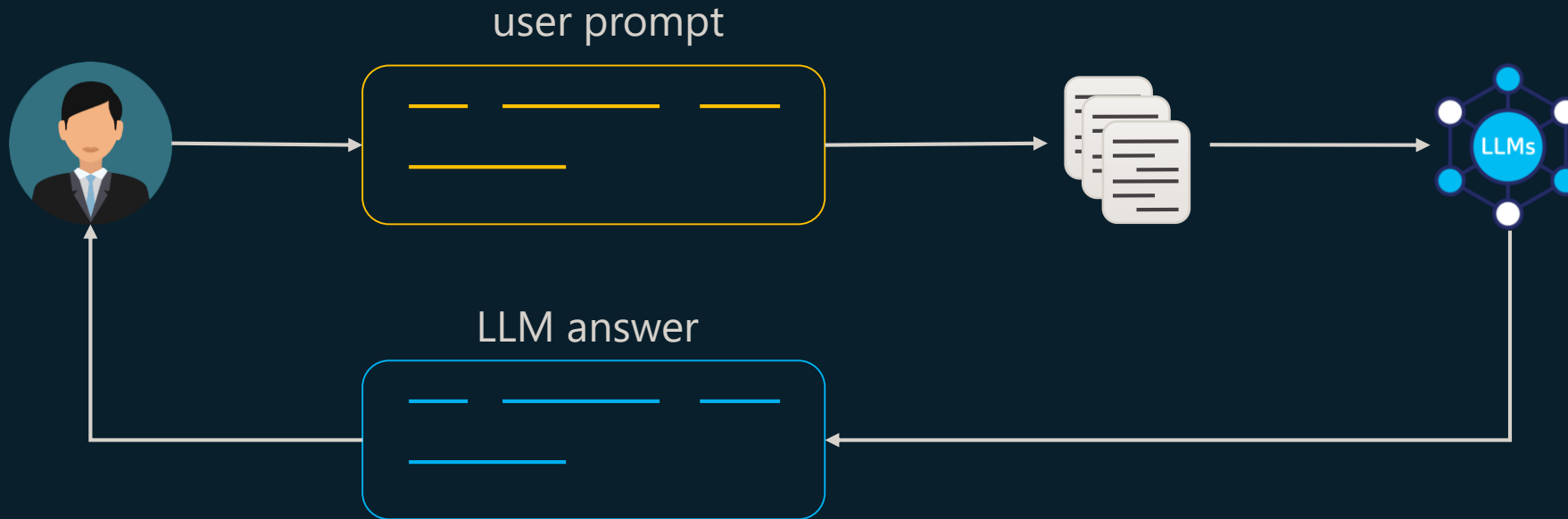
```
POST
https://{endpoint}/openai/deployments/{deployment-id}/chat/completions?api-version=2024-10-21

{
  "model": "gpt-4o",
  "messages": [
    {
      "role": "system",
      "content": "you are a helpful assistant that talks like a pirate. Generate short answer only."
    },
    {
      "role": "user",
      "content": "can you tell me how to care for a parrot?"
    }
  ],
  "temperature": 0.7,
  "stream": false,
  "max_tokens": 2000
}
```

Response

```
{
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "Aye! Provide fresh food n' water, a roomy cage, toys fer enrichment, regular vet check-ups, and plenty o' social interaction. Arr!",
        "role": "assistant"
      }
    }
  ],
  "created": 1740803441,
  "id": "chatcmpl-B68mnFNQafMCjwosWz8zPS1ZtcRV0",
  "model": "gpt-4o-2024-05-13",
  "object": "chat.completion",
  "system_fingerprint": "fp_65792305e4",
  "usage": {
    "completion_tokens": 32,
    "prompt_tokens": 39,
    "total_tokens": 71
  }
}
```

LLM Chat with RAG



Relevant answers, but limited to data sources

Microsoft

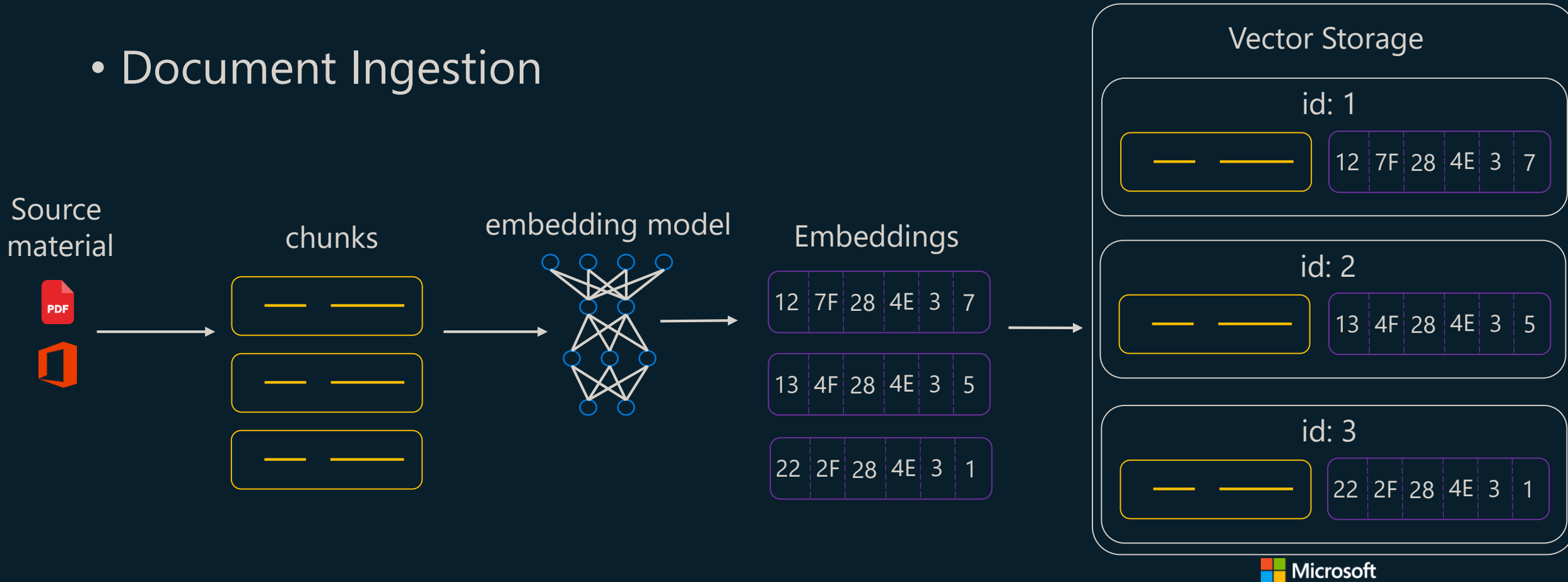
Reactor

How RAG Works (Retrieval-Augmented Generation)

- Document ingestion
- Transform, the user query into a vector
- Search in vector DB for relevant documents
- Add found documents to the context

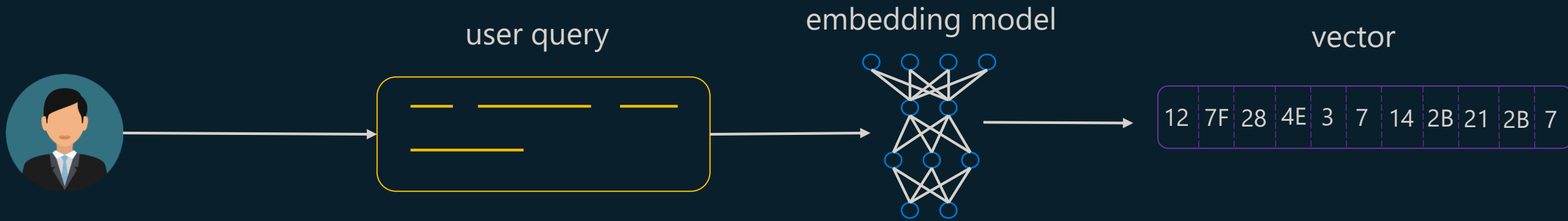
RAG (Retrieval-Augmented Generation)

- Document Ingestion



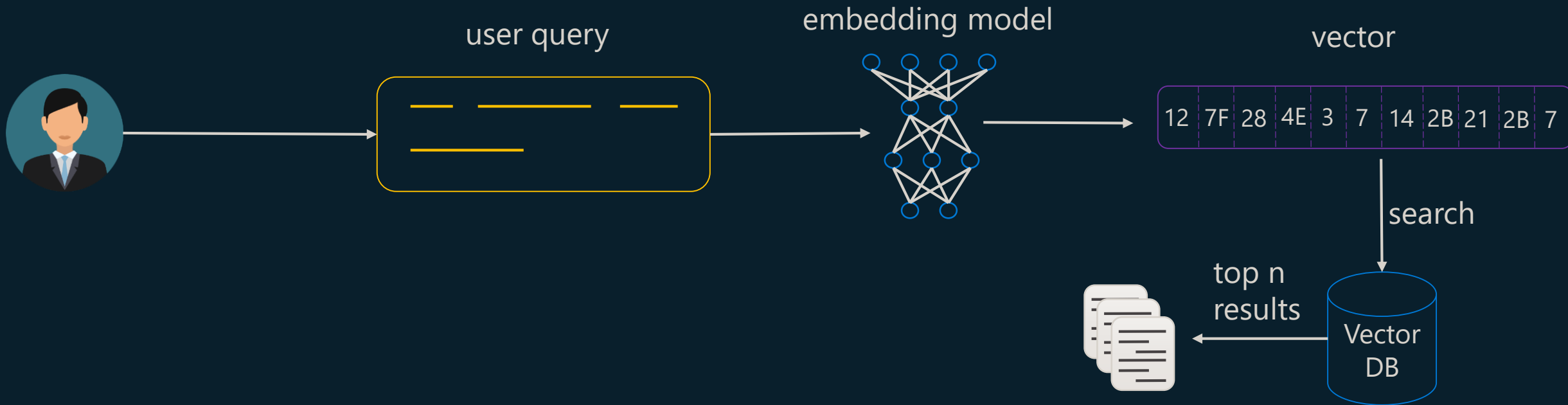
RAG (Retrieval-Augmented Generation)

- Transform, the user query into a vector



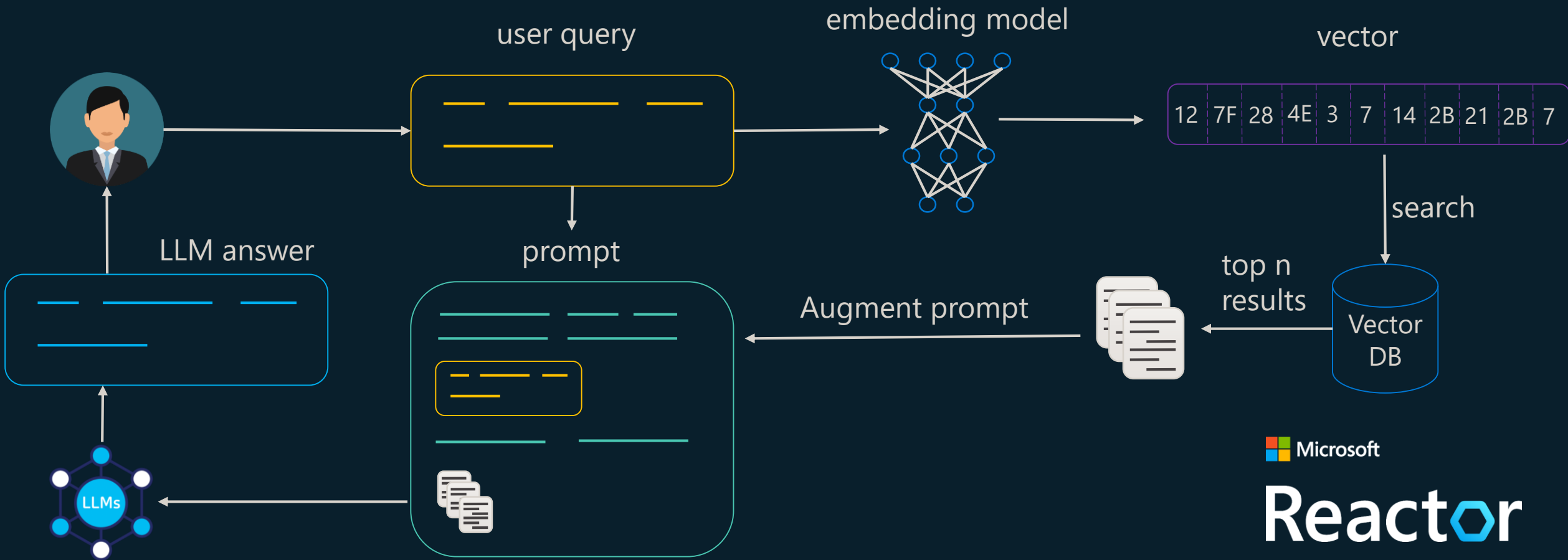
RAG (Retrieval-Augmented Generation)

- Search in vector DB for relevant documents



RAG (Retrieval-Augmented Generation)

- Add found documents to the context



AI Agent

An Agent is a semi-autonomous software that leverages large language models to operate independently over extended periods, using various tools to accomplish complex task.

It can be defined as a prescriptive implementation that follows predefined workflows.

All of these variations are considered agentic systems.

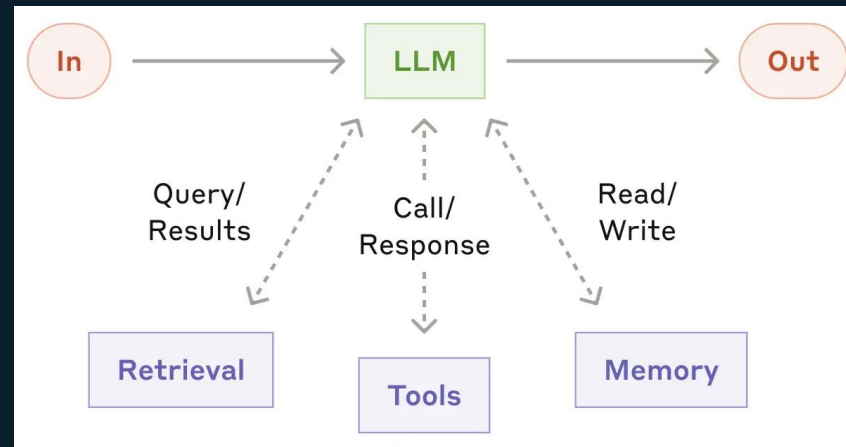
Agentic systems



Reactor

Building block

The basic building block of agentic systems is an LLM enhanced with augmentations such as retrieval, tools, and memory.

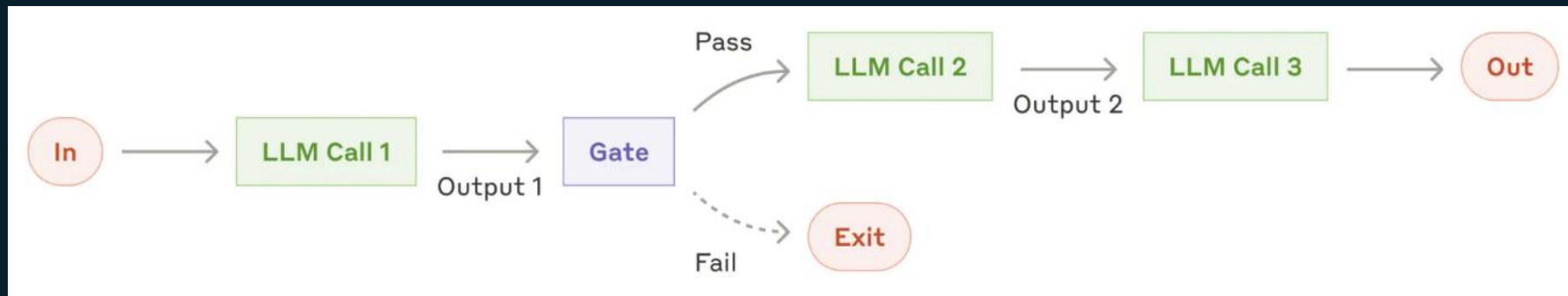


e.g.

Get weather from an API tools.

Chain Workflow

The Chain Workflow pattern exemplifies the principle of breaking down complex tasks into simpler, more manageable steps.



e.g.

Generate a marketing copy, then translating it into a different language

Parallelization Workflow

LLMs can work simultaneously on tasks and have their outputs aggregated programmatically.

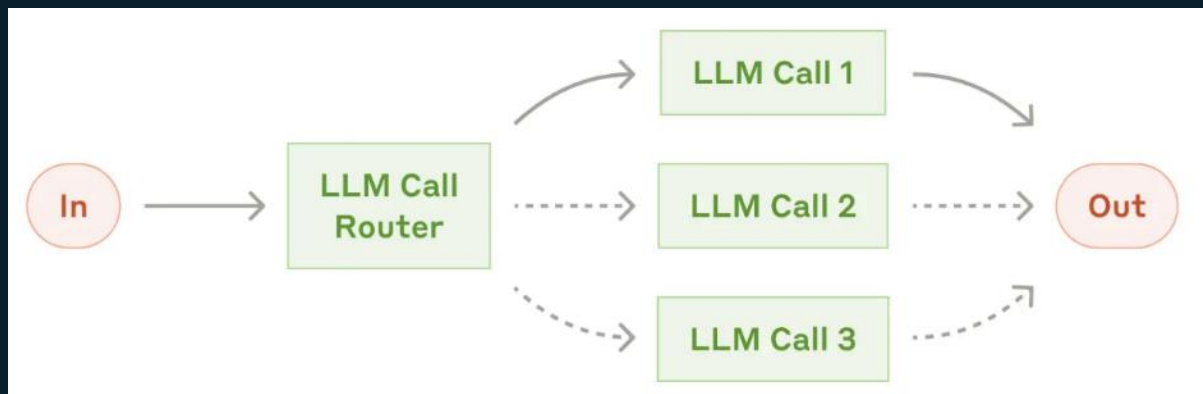


e.g.

Review code with different profile: Security,
Performance, Quality

Routing Workflow

The Routing pattern implements intelligent task distribution, enabling specialized handling for different types of input.



e.g.

Switch between LLM like for coding, reasoning, image generation, lower cost LLM

Orchestrator-Workers Workflow

In the orchestrator-workers workflow, a central LLM dynamically breaks down tasks, delegates them to worker LLMs, and synthesizes their results.



e.g.

Search data from database, web, LLM and merge them

Evaluator-Optimizer

The Evaluator-Optimizer pattern implements a dual-LLM process where one model generates responses while another provides evaluation and feedback in an iterative loop, similar to a human writer's refinement process.



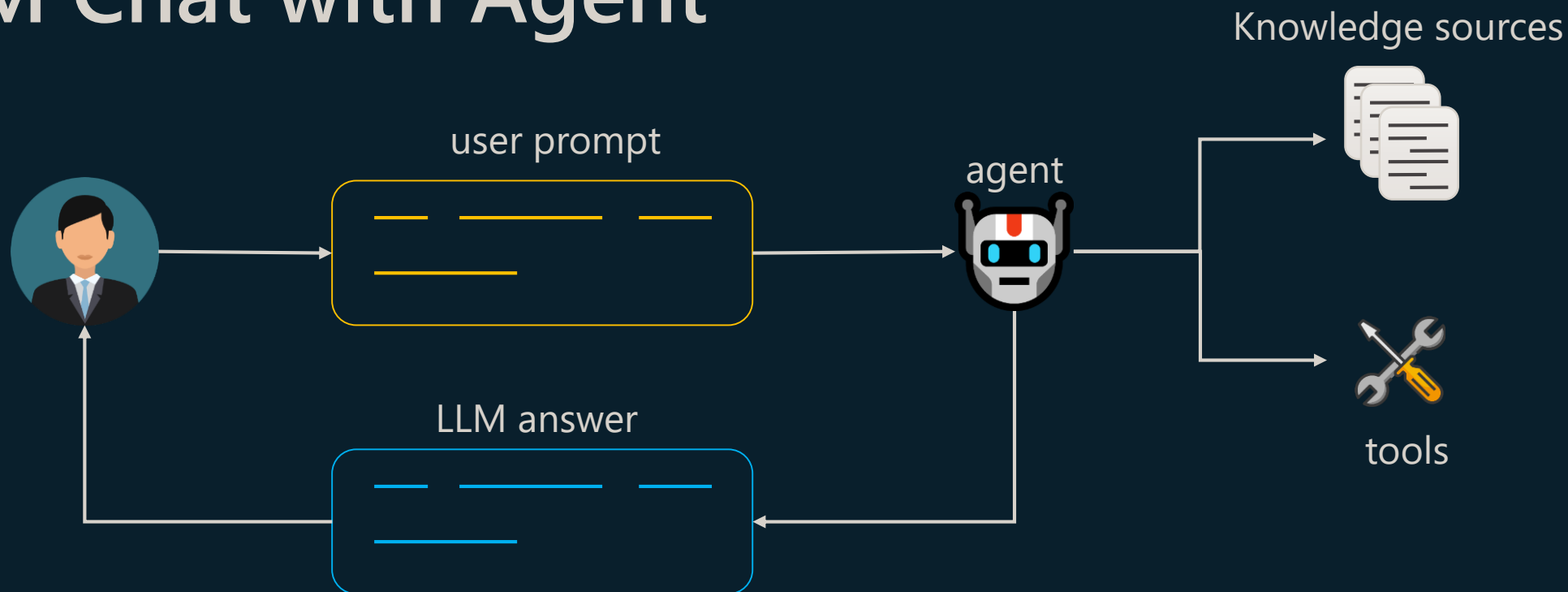
e.g.

Evaluate translation result

Microsoft

Reactor

LLM Chat with Agent

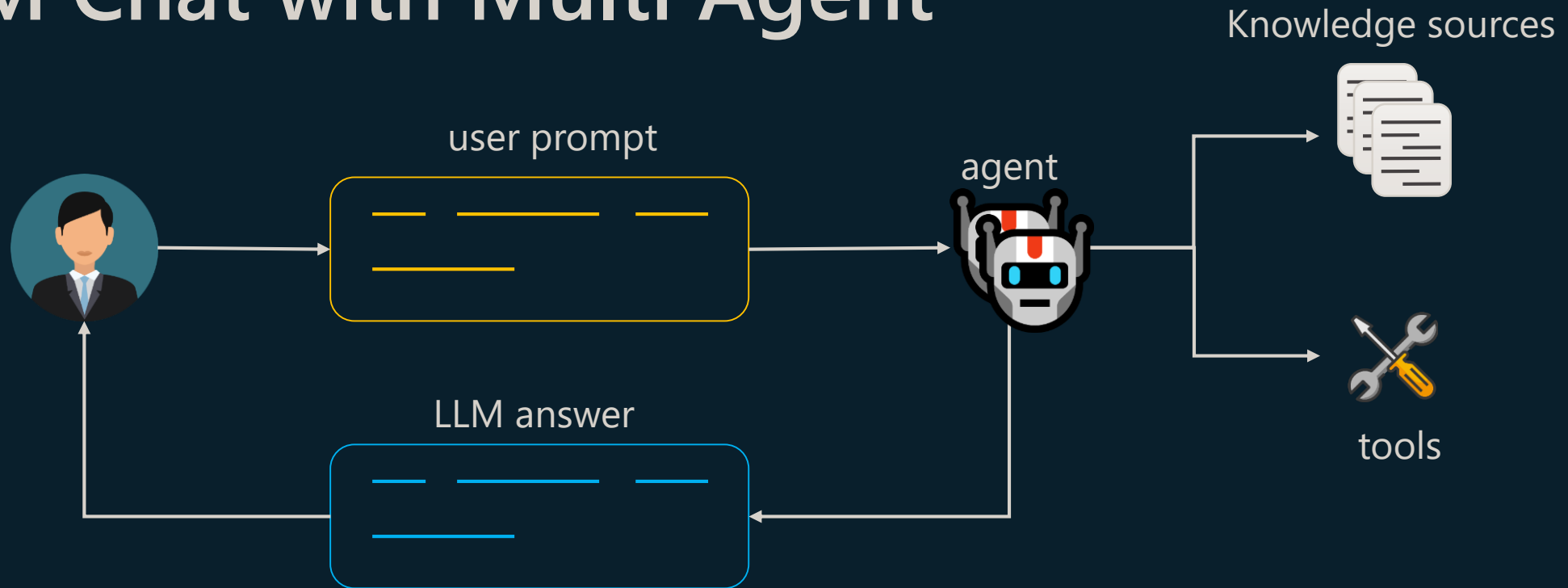


Agent perform complex tasks

Microsoft

Reactor

LLM Chat with Multi Agent



Specific assigned task agent

Microsoft

Reactor

AI Agent Service in Action

Step 1
Create an Agent

Step 2
Create a Thread

Step 3
Run the Agent

Step 4
Check the Run status

Step 5
Display the
Agent's Response

 **Agent**
Microsoft Sales Agent

Instruction
You are an advanced sales analyses agent for Microsoft, specializing in assisting users with sales data inquiries

Model


Your data (optional)
 Azure AI Search
 Files (local or Azure Blob)

Tools (optional)
File Search
Code Interpreter
Function Calling
Bing Search
Microsoft SharePoint
Microsoft Fabric
Azure Logic Apps
Azure Functions
OpenAI 3.0 specified tools

Thread
Sales analysis

 **User's message**
Tell me the total sales by region

 **Agent's message**
Here is the sales:
Europe: \$1923
America: \$1776

 **User's message**
Show a pie chart

 **Agent's message**


Run 1

1 Function Calling Tool
Query SQLite DB

2 Create message

Run 2

1 Code Interpreter Tool
Create a pie chart

2 Create message

SDK

Language	SDK
Java	Azure SDK for Java, Semantic Kernel, Spring AI, LangChain4j
Python	Azure SDK for Python, Semantic Kernel, AutoGen, LangChain
C#	Azure SDK for .NET, Semantic Kernel
JS	Azure SDK for Js, LangChainJs, Vercel AI SDK



Microsoft

Reactor

Demo

Agent



Is RAG an Agent ?

Resources

- <https://github.com/cihancinar/nextjs-azure-ai-starter>
- <https://aka.ms/Apr7AzureAIServicesOpenAI1>
- <https://aka.ms/Apr7AIFoundryConceptsRAG1>
- <https://github.com/Azure-Samples/>
- <https://github.com/Azure/GPT-RAG>
- <https://www.anthropic.com/engineering/building-effective-agents>



Reactor

Thank You

Linkedin: @cihancinar

Github: @cihancinar

X: @Cihan_Cinar_

<https://blog.cihan-cinar.com/>

Cihan Cinar

Software & AI Architect

